IN THE CLAIMS

Please cancel claims 9 and 20 amend claims 1-8, 10, 12-19, 21, 23, and 24 and add new claims 26-28 as follows:

1. (CURRENTLY AMENDED) An apparatus comprising visual display means, processing means, storage means and memory means; wherein said memory means is configured to store program instructions for updating data in a database, having persistent copies of objects that control processing steps, wherein;

a database application makes modifications to transient copies of said persistent objects;

a database thread generates database transaction requests for updating the persistent copy of the modified object in response to said modifications; and

said requests are processed, in a queue of database transaction requests, at a lower priority than said modifications.

2. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein said database is stored locally or distributed over a network to remote nodes[[;]].

3. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein said database is transaction-oriented[[;]].

4. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein said database thread includes an object cache manager[[;]].

5. (CURRENTLY AMENDED) An apparatus to claim 4, wherein said object cache manager creates said transient copies in a transient object cache according to permission from a Permit Manager[[;]].

6. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein said modifications to transient copies of said persistent objects are amendments implemented locally or remotely on said transient copies[[;]].

7. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein transient objects are stored in the main memory of a local or remote database client system or a plurality thereof[[;]].

8. (CURRENTLY AMENDED) An apparatus according to claim 1, wherein said database thread is a low priority thread[[;]].

9. (CANCELED)

10. (CURRENTLY AMENDED) An apparatus according to claim 91, wherein said database thread identifies and then executes said transactions requests asynchronously[[;]].

11. (ORIGINAL) An apparatus according to claim 1, wherein said queued transactions requests are removed from said database request queue once the said database transaction they respectively define is accomplished.

12. (CURRENTLY AMENDED) A method of updating data in a database~~, having persistent copies of objects that control processing steps, wherein~~comprising:

a database application makinges modifications to transient copies of ~~said~~ persistent copies of objects, in the database, that control processing steps;

a database thread generatinges database transaction requests for updating the persistent copy of the modified object in response to said modifications; and

processing said database transaction requests, in a queue of database transaction requests, ~~are processed~~ at a lower priority than said modifications.

13. (CURRENTLY AMENDED) A method according to claim 12, wherein said database is stored locally or distributed over a network to remote nodes[[;]].

14. (CURRENTLY AMENDED) A method according to claim 12, wherein said database is transaction-oriented[[;]].

15. (CURRENTLY AMENDED) A method according to claim 12, wherein said database thread includes an object cache manager[[;]].

16. (CURRENTLY AMENDED) A method according to claim 15, wherein said object cache manager creates said transient copies in a transient object cache according to permission from a Permit Manager[[;]].

17. (CURRENTLY AMENDED) A method according to claim 12, wherein said modifications to transient copies of said persistent objects are amendments implemented locally or remotely on said transient copies[[;]].

18. (CURRENTLY AMENDED) A method according to claim 12, wherein transient objects are stored in the main memory of a local or remote database client system or a plurality thereof[[;]].

19. (CURRENTLY AMENDED) A method according to claim 12, wherein said database thread is a low priority thread[[;]].

20. (CANCELED)

21. (CURRENTLY AMENDED) A method according to claim ~~20~~12, wherein said database thread identifies and then executes said transactions requests asynchronously[[;]].

22. (ORIGINAL) A method according to claim 12, wherein said queued transactions requests are removed from said database request queue once the said database transaction they respectively define is accomplished.

23. (CURRENTLY AMENDED) A computer-readable medium having computer-readable instructions executable by a computer such that, when executing said instructions, a computer will perform the steps of:

making modifications to transient copies of persistent objects that control processing steps;

generating database transaction requests for updating the persistent copy of the modified object in response to said modifications; and

processing said requests, in a queue of database transaction requests, at a lower priority than said modifications.

24. (CURRENTLY AMENDED) A computer-readable memory system having computer-readable data stored therein, comprising:

transient copies of persistent objects that control processing steps;

a database thread defining successive data updating processes;

a database request queue for the purpose of indexing said successive data updating processes; and

program instructions to implement said data updating processes.

25. (ORIGINAL) A computer-readable memory system according to claim 24, wherein said program instructions are configured to update objects in a database which has persistent copies of objects that control processing steps.

26. (NEW) The apparatus of claim 1 wherein when the transient copy of the persistent object in one client is accessed, any previously existing transient copy of the persistent object in another client is unloaded from transient object cache of the other client.

27. (NEW) The method of claim 12 wherein when the transient copy of the persistent object in one client is accessed, any previously existing transient copy of the persistent object in another client is unloaded from transient object cache of the other client.

28. (NEW) The computer-readable medium method of claim 23 wherein when the transient copy of the persistent object in one client is accessed, any previously existing transient copy of the persistent object in another client is unloaded from transient object cache of the other client.